UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,515	10/24/2003	Ashish Shah	MSFT-2844/306723.01	9293

41505 7590 06/18/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
SAEED, USMAAN	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
06/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/692,515	SHAH ET AL.	
	Examiner	Art Unit	
	USMAAN SAEED	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 April 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Receipt of Applicant's Amendment, filed 04/10/2009 is acknowledged.

Claims 1-30 are pending in this office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Multer et al.** (**Multer** hereinafter) (U.S. Patent No. 6,694,336) in view of **Herbert P. Sutter.** (**Sutter** hereinafter) (U.S. Patent No. 6,446,092), further in view of **Pham et al.** (**Pham** hereinafter) (U.S. PG Pub No. 2004/0078568).

With respect to claim 1, **Multer** teaches **a storage platform system including a processor and a computer readable medium, said storage system comprising:**

"instructions for an operating system, the operating system including a kernel, wherein the kernel includes a database management system integrated with a file system, the database management program integrated with the file system configured to store data in the file system as file streams, generate items that include metadata for the file streams and store the item in the database management program" as (**Multer** Col 5, Lines 9-16, figure 10).

"the operating system including a synchronization subsystem configured to synchronize the data stored in the database management program integrated with file system with a remote computer based on changes that are sequentially enumerated and tracked on a per change unit basis" as items such as when to sync, how to sync, trigger the delta module 950 to perform a synchronization operation (**Multer** Col 11, Lines 55-57). The invention, roughly described, comprises a difference information receiver, a difference information transmitter and a difference information synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25). EnumItem interface allows the enumeration of either Folder objects or Item objects or both (**Multer** Col 20, Lines 16-18).

Multer teaches **smallest change unit** as if a single bit on a system changes, the system of the present invention allows synchronization of that bit on another system.

Changes are described as a sequence of bite-level change operations but does not explicitly teaches **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program," "a base schema and a mechanism configured to extend the base schema to define a schema for the data" and "based on the schema for the data, wherein a change unit is a smallest piece of schema that is individually tracked by each instance of the storage platform and the size of a change unit is adjustable."**

However, **Sutter** discloses **"a base schema and a mechanism configured to extend the base schema to define a schema for the data"** as the application database 34, direction for the database schema is defined as "up" for the direction of the "one" end of all one-to-many relationships and "down" as the direction of the "many" end. Accordingly, an activity comprises the activity record and some or all related records beneath it (i.e. "down") in the schema (**Sutter** Col 37, Lines 7-11). Further the extended schema for the application database 100 is shown in FIG. 11 (**Sutter** Col 40, Lines 13-15).

"based on the schema for the data, wherein a change unit is a smallest piece of schema that is individually tracked by each instance of the storage platform and the size of a change unit is adjustable" as With record-level granularity, the fields of an entire record are replicated together, and with field-level granularity, each field in a record is replicated separately, which solves the false collisions of record-level replication (**Sutter** Col 2, Lines 63-64 and Col 3, lines 4-6).

Sutter further teaches second level of replication control is where record fragment is the unit of replication. The record fragment defines the granularity with which changes propagate through the distributed system. Some columns in a table will have a common update responsibility, and grouping these columns together in fragments allows designers to achieve full replication control without having to micro-manage replication rules on a field-level basis (**Sutter** Col 76, Lines 9-16).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Sutter's** teachings would have allowed **Multer** to provide efficient replication by providing timestamps which are used to calculate the age of the change units and timestamp also speed up the replication process.

Multer and Sutter teach the elements of claim 1 as noted above but do not explicitly teach **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program."**

However, **Pham** discloses **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program"** as the platform 14 conventionally includes an operating system kernel 42 supporting execution of applications, such as a database management system (DBMS) 44 and other server applications 46, in a user mode execution space. The operating system kernel 42 also preferably supports execution of an authentication agent program 48 substantially, if not completely, within a

kernel mode execution space. A virtual file system switch (VFS) 50 provides a conventional interface to any number of different conventional file systems (xFS) 52 as necessary to access conventional direct attached storage 18 and, for example, storage devices 54 (**Pham** Abstract and Paragraph 0036).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Pham's** teachings would have allowed **Multer and Sutter** to provide an efficient and effective mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain.

With respect to claim 2, **Multer** teaches **"the system of claim 1 wherein the synchronization subsystem synchronizes only a subset of data, from among the entirety of data on said data store, during a synchronization operation"** as generating first difference information upon a change to the data files by comparing the change to the data store; receiving second difference information for a subset of said data files from a second system; and applying said difference information to said subset of said data files (**Multer** Abstract).

With respect to claim 3 and 4, **Multer and Sutter** do not explicitly teach **"wherein instruction that effectuate the database management program integrated with the fie system are configured to execute during kernel mode" and "wherein the operating system further includes an application program interface**

that is configured to execute during kernel mode, wherein the application program interface that is configured to expose the items stored in the database management program to applications executing in user mode of the operating system."

However, **Pham** discloses **"wherein instruction that effectuate the database management program integrated with the fie system are configured to execute during kernel mode"** as (Pham Abstract Paragraphs 0013 and 0036) **"wherein the operating system further includes an application program interface that is configured to execute during kernel mode, wherein the application program interface that is configured to expose the items stored in the database management program to applications executing in user mode of the operating system"** as (Pham Paragraphs 0008, 0036-0037 and 0049).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Pham's** teachings would have allowed **Multer and Sutter** to provide an efficient and effective mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain.

With respect to claim 5, **Multer** teaches **"wherein the synchronization subsystem is configured to synchronize changes independent of the remote computer system"** as the invention, roughly described, comprises a difference information receiver, a difference information transmitter and a difference information

synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25 and Figures 1-5).

With respect to claim 6, **Multer** teaches **"the system of claim 1 wherein conflicts in synchronization are automatically detected and resolved based on predefined determinable criteria"** as in this embodiment, storage server 300 may include routines, described below, for resolving conflicts between data which has changed on both System A and System B independently after the last point in times when the systems were synchronized (**Multer** Col 7, Lines 1-5).

With respect to claim 7, **Multer** teaches **"the system of claim 6 wherein certain of said conflicts are resolved by being logged for manual resolution by an end-user"** as if both files have changed, then the synchronization routine presents the option of conflict resolution to the user (**Multer** Col 2, Lines 39-41).

With respect to claim 8, **Multer** teaches **"the system of claim 1 wherein the synchronization subsystem tracks the state of previous synchronizations with a sync partner, and thereby only synchronizes change units with that partner that have changed since the last synchronization"** as the system includes: a system data store associated with the processing device including a representation of a previous state of application data in the application data store; a difference engine generating

difference information associated with a change to said application data store; and an application interface, interpreting application data for the difference engine. The difference engine may further comprise a delta engine comparing the change to said application data store to said system data store to construct difference information (**Multer** Abstract).

With respect to claim 9, **Multer** teaches **a method for synchronizing data stored in a computer system, said method comprising:**

"executing an operating system that includes a kernel, the kernel includes a database management program integrated with a file system, storing by the database management program integrated with the file system data in the file system as file streams, generating by the database management program integrated with the file system, items that include metadata for the file streams, wherein metadata is defined by the schema for the data and storing the item in the database management program" as (**Multer** Col 5, Lines 9-16, figure 10).

"dividing said data into programmably defined, change units" as each device engine performs mapping and translation steps necessary for applying the data packages to the local format required for that type of information in the application data stores 822-828 (**Multer** Col 11, Lines 11-14). In one embodiment, the invention comprises a set of programs specifically designed to transmit and/or receive differencing data from one device to another device, irrespective of the type of file system, data, content, or system hardware configuration (**Multer** Col 5, Lines 17-16-20).

The objects in universal data format are device, (application) data class, store, folder, item, and data fields (**Multer** Col 18, Lines 40-41).

Examiner interprets the data stores 822-828 as multiple instances of a storage platform/file system and each instance/data store has folders, items, data fields which are interpreted as change units by the examiner.

"sequentially enumerating changes to said data and tracking said changes on a per change unit basis" as items such as when to sync, how to sync, trigger the delta module 950 to perform a synchronization operation (**Multer** Col 11, Lines 55-57). The invention, roughly described, comprises a difference information receiver, a difference information transmitter and a difference information synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25). EnumItem interface allows the enumeration of either Folder objects or Item objects or both (**Multer** Col 20, Lines 16-18).

"tracking changes to the database management program as well change to a remote computer system in sync community" as the system includes: a system data store associated with the processing device including a representation of a previous state of application data in the application data store; a difference engine generating difference information associated with a change to said application data store; and an application interface, interpreting application data for the difference engine. The difference engine may further comprise a delta engine comparing the

change to said application data store to said system data store to construct difference information (**Multer** Abstract).

"identifying new changes by comparing the enumerated changes to said data to the changes of the remote computer system" as the system includes: a system data store associated with the processing device including a representation of a previous state of application data in the application data store; a difference engine generating difference information associated with a change to said application data store; and an application interface, interpreting application data for the difference engine. The difference engine may further comprise a delta engine comparing the change to said application data store to said system data store to construct difference information (**Multer** Abstract). The invention, roughly described, comprises a difference information receiver, a difference information transmitter and a difference information synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25 and Figures 1-5).

Multer teaches **smallest change unit** as if a single bit on a system changes, the system of the present invention allows synchronization of that bit on another system. Changes are described as a sequence of bite-level change operations but does not explicitly teaches **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program," "a base schema and a mechanism configured to extend the base schema to define a schema for the data" and**

"based on the schema for the data, wherein a change unit is a smallest piece of schema that is individually tracked by each instance of the storage platform and the size of a change unit is adjustable."

However, **Sutter** discloses **"each instance of the storage platform including a base schema and a mechanism configured to extend the base schema to define a schema for the data"** as the application database 34, direction for the database schema is defined as "up" for the direction of the "one" end of all one-to-many relationships and "down" as the direction of the "many" end. Accordingly, an activity comprises the activity record and some or all related records beneath it (i.e. "down") in the schema (**Sutter** Col 37, Lines 7-11). Further the extended schema for the application database 100 is shown in FIG. 11 (**Sutter** Col 40, Lines 13-15).

"based on the schema for the data, wherein a change unit is a smallest piece of schema that is individually tracked by each instance of the storage platform and the size of a change unit is adjustable" as With record-level granularity, the fields of an entire record are replicated together, and with field-level granularity, each field in a record is replicated separately, which solves the false collisions of record-level replication (**Sutter** Col 2, Lines 63-64 and Col 3, lines 4-6).

Sutter further teaches second level of replication control is where record fragment is the unit of replication. The record fragment defines the granularity with which changes propagate through the distributed system. Some columns in a table will have a common update responsibility, and grouping these columns together in

fragments allows designers to achieve full replication control without having to micro-manage replication rules on a field-level basis (**Sutter** Col 76, Lines 9-16).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Sutter's** teachings would have allowed **Multer** to provide efficient replication by providing timestamps which are used to calculate the age of the change units and timestamp also speed up the replication process.

Multer and Sutter teach the elements of claim 9 as noted above but do not explicitly teach **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program."**

However, **Pham** discloses **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program"** as the platform 14 conventionally includes an operating system kernel 42 supporting execution of applications, such as a database management system (DBMS) 44 and other server applications 46, in a user mode execution space. The operating system kernel 42 also preferably supports execution of an authentication agent program 48 substantially, if not completely, within a kernel mode execution space. A virtual file system switch (VFS) 50 provides a conventional interface to any number of different conventional file systems (xFS) 52 as necessary to access conventional direct attached storage 18 and, for example, storage devices 54 (**Pham** Abstract and Paragraph 0036).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Pham's** teachings would have allowed **Multer and Sutter** to provide an efficient and effective mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain.

With respect to claim 11, **Multer** teaches **"detecting synchronization conflicts at the level of change unit granularity"** as in this embodiment, storage server 300 may include routines, described below, for resolving conflicts between data which has changed on both System A and System B independently after the last point in times when the systems were synchronized (**Multer** Col 7, Lines 1-5).

With respect to claim 12, **Multer** teaches **"the method of claim 10, further comprising: instances reporting success, failure, and/or conflicts at individual change unit level on change application, the instance comprising sync data"** as in this embodiment, storage server 300 may include routines, described below, for resolving conflicts between data which has changed on both System A and System B independently after the last point in times when the systems were synchronized (**Multer** Col 7, Lines 1-5).

"applications using sync data for updating a backend state" as items such as when to sync, how to sync, trigger the delta module 950 to perform a synchronization operation (**Multer** Col 11, Lines 55-57). The invention, roughly described, comprises a

difference information receiver, a difference information transmitter and a difference information synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25).

With respect to claim 13, **Multer** teaches **a method for synchronizing data said method comprising:**

"executing an operating system that includes a kernel, the kernel includes a database management program integrated with a file system, storing by the database management program integrated with the file system data in the file system as file streams, generating by the database management program integrated with the file system, items that include metadata for the file streams, wherein metadata is defined by the schema for the data and storing the item in the database management program" as (**Multer** Col 5, Lines 9-16, figure 10).

"receiving by an application program interface exposed to a shell of the operating system updated state information for said remote computer system that, reflect new changes that have been made since the last synchronization" as the system includes: a system data store associated with the processing device including a representation of a previous state of application data in the application data store; a difference engine generating difference information associated with a change to said application data store; and an application interface, interpreting application data for the difference engine. The difference engine may further comprise a delta engine

comparing the change to said application data store to said system data store to construct difference information (**Multer** Abstract and figures 1-5).

"applying a conflict resolution policy selected from a plurality of conflict resolution policies, and tracking success or failure for each change on a change unit by change unit basis" as conflict resolution module 940 (**Multer** Figure 9A). In this embodiment, storage server 300 may include routines, described below, for resolving conflicts between data which has changed on both System A and System B independently after the last point in times when the systems were synchronized (**Multer** Col 7, Lines 1-5).

"modifying the item in accordance with new changes that have been made to the change unit of the item and a conflict resolution policy selected from the plurality of conflict resolution policies" as (Multer Col 17, Lines 51-62 and Col 7, Lines 1-5).

"wherein changes are sequentially enumerated and tracked on a per change unit basis" as items such as when to sync, how to sync, trigger the delta module 950 to perform a synchronization operation (**Multer** Col 11, Lines 55-57). The invention, roughly described, comprises a difference information receiver, a difference information transmitter and a difference information synchronizer which cooperate in a system or device to update data in the device with data received from other systems, or provide data for other systems to use in updating themselves (**Multer** Col 3, Lines 21-25). EnumItem interface allows the enumeration of either Folder objects or Item objects or both (**Multer** Col 20, Lines 16-18).

"storing data that reflects the new changes in the file system as file streams" as (Multer Abstract).

Multer teaches **smallest change unit** as if a single bit on a system changes, the system of the present invention allows synchronization of that bit on another system. Changes are described as a sequence of bite-level change operations but does not explicitly teaches **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program," "wherein the item conforms to a schema derived from a base schema, the schema defining the size of a change unit, wherein change unit being a smallest piece of schema that is individually tracked."**

However, **Sutter** discloses **"wherein the item conforms to a schema derived from a base schema, the schema defining the size of a change unit, wherein change unit being a smallest piece of schema that is individually tracked"** as the application database 34, direction for the database schema is defined as "up" for the direction of the "one" end of all one-to-many relationships and "down" as the direction of the "many" end. Accordingly, an activity comprises the activity record and some or all related records beneath it (i.e. "down") in the schema (**Sutter** Col 37, Lines 7-11). Further the extended schema for the application database 100 is shown in FIG. 11 (**Sutter** Col 40, Lines 13-15).

With record-level granularity, the fields of an entire record are replicated together, and with field-level granularity, each field in a record is replicated separately, which

solves the false collisions of record-level replication (**Sutter** Col 2, Lines 63-64 and Col 3, lines 4-6).

Sutter further teaches second level of replication control is where record fragment is the unit of replication. The record fragment defines the granularity with which changes propagate through the distributed system. Some columns in a table will have a common update responsibility, and grouping these columns together in fragments allows designers to achieve full replication control without having to micro-manage replication rules on a field-level basis (**Sutter** Col 76, Lines 9-16).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Sutter's** teachings would have allowed **Multer** to provide efficient replication by providing timestamps which are used to calculate the age of the change units and timestamp also speed up the replication process.

Multer and Sutter teach the elements of claim 13 as noted above but do not explicitly teach **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program."**

However, **Pham** discloses **"a database management program integrated with a file system configured to store data in the file system as file streams, and store the item in the database management program"** as the platform 14 conventionally includes an operating system kernel 42 supporting execution of applications, such as a database management system (DBMS) 44 and other server applications 46, in a user

mode execution space. The operating system kernel 42 also preferably supports execution of an authentication agent program 48 substantially, if not completely, within a kernel mode execution space. A virtual file system switch (VFS) 50 provides a conventional interface to any number of different conventional file systems (xFS) 52 as necessary to access conventional direct attached storage 18 and, for example, storage devices 54 (**Pham** Abstract and Paragraph 0036).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Pham's** teachings would have allowed **Multer and Sutter** to provide an efficient and effective mechanism for reliably securing persistent data in a manner eminently subject to cooperative management and control within a security domain.

With respect to claim 14, **Multer** teaches **"the method of claim 13, further comprising: calculating a new state of the data source based on the success or failure for each change on a change unit by change unit basis, storing the new state information, and transmitting the new state information to the remote computer system"** as the system includes: a system data store associated with the processing device including a representation of a previous state of application data in the application data store; a difference engine generating difference information associated with a change to said application data store; and an application interface, interpreting application data for the difference engine. The difference engine may further comprise a delta engine comparing the change to said application data store to

said system data store to construct difference information. In a further aspect, a method for updating data files in a first system is provided. The method includes the steps of providing a data store associated with the first system and including information representing data in the data files at a previous time state; generating first difference information upon a change to the data files by comparing the change to the data store; receiving second difference information for a subset of said data files from a second system; and applying said difference information to said subset of said data files (**Multer** Abstract).

With respect to claim 15, **Multer** teaches **the method of claim 13 further comprising: "transmitting to the remote computer system the success or failure for each change on a change unit by change unit basis"** as in this embodiment, storage server 300 may include routines, described below, for resolving conflicts between data which has changed on both System A and System B independently after the last point in times when the systems were synchronized (**Multer** Col 7, Lines 1-5). Examiner interprets conflicts as failure or success.

"said hardware/software interface system of the replica calculating a new state information for the data source based on the success or failure for each change to the data source on a change unit by change unit basis" as once the engine server lock is acquired, the storage server will be checked to determine whether a new version of the data exists on the storage server at step 1430. If no new version exists, the synchronization process ends. If a new version of the data exists, the device

engine will retrieve the difference information at step 1435 "to get .DELTA.." Once a .DELTA. is retrieved, conflicts are resolved at step 1450. The resolve conflicts step allows a user to resolve conflicts to multiple types of data, which have been changed on both the server portion of the device and in the local data (**Multer** Col 37, Lines 3-13).

"receiving new state information that reflects whether each change was successful from the remote computer system; and storing said new state information" as (**Multer** Col 12, Lines 39-53). Examiner interprets that versioning module keeps track of the new and old states by assigning a universal unique ID.

With respect to claim 19, **Multer** teaches **"wherein instruction for the operating shell include instruction for receiving changes to items from a remote computer system and the instructions for the shell configured to transmit changes to items to the application program interface"** as (Multer Col 5, Lines 56-67 to Col 6, Lines 1-5 and Col 12, Lines 28-38).

Claims 10-12, 16-18, 20-27, and 28-30 are essentially the same as claims 1-9, and 13-15 except they set forth the claimed invention as a computer-readable medium comprising instructions and are rejected for the same reason as applied hereinabove.

Response to Arguments

3. Applicant's arguments filed 04/10/2009 have been fully considered but they are not persuasive.

Applicant argues that **Pham** does not teach or suggest **"the operating system including a kernel, wherein the kernel includes a database management program integrated with a file system."**

In response to the preceding arguments, first examiner respectfully submits that Microsoft's computer dictionary defines Kernel as the core of an operating system - the portion of the system that manages memory, files, and peripheral devices; maintains the time and data; launches applications; and allocates system resources.

Further, examiner respectfully submits that Pham teaches **"the operating system including a kernel, wherein the kernel includes a database management program integrated with a file system"** as the platform 14 conventionally includes an operating system kernel 42 supporting execution of applications, such as a database management system (DBMS) 44 and other server applications 46, in a user mode execution space. The operating system kernel 42 also preferably supports execution of an authentication agent program 48 substantially, if not completely, within a kernel mode execution space. A virtual file system switch (VFS) 50 provides a conventional interface to any number of different conventional file systems (xFS) 52 as necessary to access conventional direct attached storage 18 and, for example, storage devices 54 (**Pham** Abstract and Paragraph 0036, 0032 and figures 1 and 2).

In these paragraphs and figures, Pham teaches file system 26 and operating system and applications 24. This operating system and application are being integrated with the file system through a security interposer layer 22. Therefore the operating

system containing kernel and applications such as database management program are being integrated with a file system by the use of a security interposer layer. Examiner interprets the platform 14 as having the claimed "operating system including a kernel, wherein the kernel includes a database management program integrated with a file system."

Claims must be given the broadest reasonable interpretation during examination and limitations appearing in the specification but not recited in the claim are not read into the claim (See M.P.E.P. 2111 [R-I]).

Conclusion

4. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to USMAAN SAEED whose telephone number is (571)272-4046. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain Alam can be reached on (571)272-3978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Usmaan Saeed/
Examiner, Art Unit 2166
June 15, 2009

Usmaan Saeed
Patent Examiner
Art Unit: 2166

/Hosain T Alam/
Supervisory Patent Examiner, Art Unit 2166